Foreground-Background Distribution Modeling Transformer for Visual Object Tracking

Dawei Yang[1,*], Jianfeng He[1,*], Yinchao Ma[1], Qianjin Yu[1], Tianzhu Zhang[1,†]
[1] University of Science and Technology of China
{yangdawei,hejf,imyc,sa21010105}@mail.ustc.edu.cn, {tzzhang}@ustc.edu.cn

Abstract

Visual object tracking is a fundamental research topic with a broad range of applications. Benefiting from the rapid development of Transformer, pure Transformer trackers have achieved great progress. However, the feature learning of these Transformer-based trackers is easily disturbed by complex backgrounds. To address the above limitations, we propose a novel foreground-background distribution modeling transformer for visual object tracking (F-BDMTrack), including a fore-background agent learning (FBAL) module and a distribution-aware attention (DA^2) module in a unified transformer architecture. The proposed F-BDMTrack enjoys several merits. First, the proposed FBAL module can effectively mine fore-background information with designed fore-background agents. Second, the DA^2 module can suppress the incorrect interaction between foreground and background by modeling fore-background distribution similarities. Finally, F-BDMTrack can extract discriminative features under ever-changing tracking scenarios for more accurate target state estimation. Extensive experiments show that our F-BDMTrack outperforms previous state-of-the-art trackers on eight tracking benchmarks.

1. Introduction

Visual object tracking(VOT) aims to locate the position of a class-agnostic target in a video sequence given the target in the first frame, which is a fundamental and essential research topic in computer vision. Due to its great application potential (such as video surveillance [9, 49, 9], anti-UAV tracking [64, 25, 29], and automatic driving [47, 41, 15]), VOT has attracted substantial attention and has been developed tremendously [56, 31, 17]. However, as a video processing task, visual object tracking still faces various challenges including deformation, motion blur, and susceptibility to background interference [51, 27, 24].

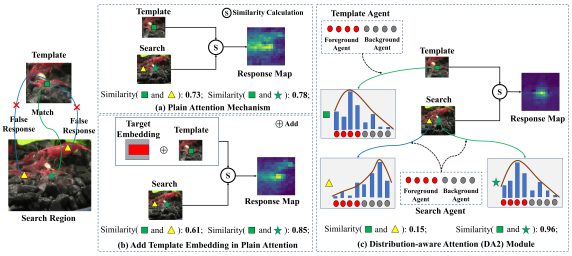

Figure 1. Response map comparisons among different attention mechanisms. Both (a) and (b) have incorrect similarities between the target (green square patch) and background distractor (yellow triangle patch) due to similar appearance, causing messy response map. Differently, in our proposed DA^2 module (c), the similarity between target and distractor is suppressed, since the similarity is obtained according to their fore-background distributions.

To deal with the above challenges, numerous approaches have been proposed [23, 2, 3, 59, 10, 5, 62]. These methods can be generally divided into three main paradigms, including CNN-based [8, 52, 58, 35, 3, 13], hybrid CNN-Transformer [7, 59, 36, 50, 42], and pure Transformer trackers [5, 62, 32, 57]. CNN-based trackers first extract features from the template and search region separately through a shared convolutional neural network (CNN), and then the target state is estimated by calculating cross-correlation [2, 58, 52] between features in template and search regions, or learning a discriminative correlation filter [12, 3, 13]. With the successful development of transformer in the computer vision field, an increasing number of hybrid CNN-Transformer trackers [7, 59, 42] have been proposed. These trackers also leverage CNN to extract features separately, but adopt transformer to realize the feature interaction between the template and search region, which can alleviate the loss of discriminative foreground information. However, for the above two paradigm trackers, there is no interaction between the template and the search region when extracting features. These target-unaware feature extraction ways have limited target-background discriminative power, especially when the target category is not seen

*Equal Contribution
†Corresponding Author

in the training dataset. To this end, pure Transformer trackers [5, 62, 32, 57] are proposed to build interaction between template and search region along with the feature extraction. Thanks to the strong representation and interaction capability of ViT/SwinT variants [14, 38, 55], pure Transformer trackers have achieved remarkable improvements.

Despite the success of the above pure Transformer trackers, the feature learning of these methods are easily disturbed by complex backgrounds due to insufficient consideration of fore-background relationship. To make pure Transformer architectures more suitable for discriminative feature learning in visual object tracking, there are two core points that need to be considered. (1) **Fore-background information mining.** Previous methods [5, 22, 21] usually mine fore-background information from the template according to the given bounding box (bbox). However, the tracking target can be continuously changing, making it difficult to learn target discriminative features for current search region, if only guided by fore-background information of the template. Therefore, it is necessary to propose a effective way to mine fore-background information for both template and search region. (2) **Target-aware feature interaction.** Most of popular Transformer trackers [62, 10, 57] adopt the plain attention mechanism to build interaction between template and search region. Since attention scores are obtained by appearance similarity, the template features will mistakenly aggregate background information and further affect features of search region during feature interaction, resulting messy response map (Figure 1 (a)). Although some methods [22, 42] introduce additional target embeddings to enhance the target information, they could still be disturbed by distractors, making features of the search region have limited target discriminative power. And the response map is also unsatisfactory (Figure 1 (b)). Therefore, it is urgent to design a new attention mechanism to achieve better target-aware feature interaction.

Motivated by the above discussion, we propose a novel Foreground-Background Distribution Modeling Transformer for visual object tracking (F-BDMTrack), which consists of a fore-background agent learning (FBAL) module and a distribution-aware attention (DA^2) module in a unified transformer architecture. **In the fore-background agent learning module**, we aim to mine foreground and background information from both the template and the search region. Specifically, we initialize a set of fore-background agents. And fore-background agents for the template (search region) are obtained by aggregating foreground and background information in the template (search region), which can be used to guide the subsequent feature interaction. **In the distribution-aware attention module**, it is proposed to realize target-aware feature interaction. Specifically, we calculate the similarity scores between each patch feature and the fore-background agents, which can be regarded as fore-background distributions for each patch feature. Then we can obtain the attention score by compute the similarity between their distributions, which are later used for feature aggregation. As shown in Figure 1 (c), foreground patches and background patches could have different fore-background distributions even though they have similar appearance. Thus, we can aggregate more useful information and learn more discriminative features.

The main contributions of this work are summarized as follows. (1) We develop a novel Foreground-background Distribution Modeling Transformer for visual object tracking (F-BDMTrack), which can extract features with high target discriminative power under ever-changing tracking scenarios. (2) The proposed FBAL module can effectively mine fore-background information for both template and search region with designed fore-background agents. And the proposed DA^2 module can effectively suppress the interaction between foreground and background, which helps learn more discriminative features for visual tracking. (3) Extensive experimental results on eight benchmarks show that our method attains state-of-the-art performance, verifying the superiority of our F-BDMTrack.

2. Related Work

Visual object tracking (VOT). Current popular trackers can be divided into CNN-based trackers [8, 2, 58, 35, 40, 12, 3, 13], hybrid CNN-Transformer trackers [59, 7, 50, 53, 42], and pure Transformer trackers [62, 5, 36, 32, 57] according to the network architecture. CNN-based trackers commonly learn features of template and search region via share-weighted convolutional neural networks (CNN), and the interaction between template and search region is realized by cross correlation [2, 58, 35] or correlation filter [12, 3, 13]. However, due to the lack of global perception and the simple interaction strategy, some target-background discriminative information may be lost, restricting the development of CNN-based trackers. Recently, Vision Transformer brings a new solution to visual tracking. Hybrid CNN-Transformer trackers reserve the extraction of features with CNNs and utilize attention mechanisms to establish global dependencies between features. Typically, STARK [59] utilizes the Tramsformer to aggregate spatial-temporal cues for target location. Moreover, ToMP [42] uses a Transformer-based architecture to encode a long-term target representation to localize the target. Nevertheless, these Hybrid CNN-Transformer trackers still extract features of template and search region separately, which causes extracted features to be unaware of the tracking target. To alleviate this limitation, pure Transformer-based trackers [62, 5, 57] are proposed. For example, OSTrack [62] and SimTrack [5] unify the feature extraction and the feature relation modeling in Vision Transformer [14]. They achieve superior performance, proving the potential of pure Transformer

architectures in visual tracking. However, these trackers [62, 5] aggregate features according to appearance similarities through plain attention mechanisms. Without considering fore-background relationship for the template and search region, target features may mistakenly aggregate background noise in the search region. Differently, in our work, we propose the distribution aware attention module to aggregate features according to their fore-background distribution similarities, which can effectively suppress incorrect interaction between foreground and background.

Fore-background exploitation in visual object tracking. There are several tracking methods [21, 42, 5, 46, 19] actively exploring ways to mine fore-background information for robust tracking. STMTrack [21] leverages foreground-background masks derived by bounding boxes of template sequences to highlight target representations. And ToMP [42] is proposed to learn foreground embeddings from a set of template sequences, which can be used to enhance target features of the template and assist in target state estimation. However, the above methods for exploiting fore-background generally use fixed foreground embedding for template images, ignoring the fact that tracking target can be continuously changing. Besides, they do not analyze the fore-background information of the search region. Furhtermore, RTS [46] utilizes an additional segmentation network to provide more precise object masks, leading to better performance improvements. And SNLT [19] adopts additional natural language labels to help extract better target embeddings for distinguishing foreground and background. These methods utilize additional network components and labels to mine fore-background information, which affects the independence of the network architecture. In contrast, our method is proposed to fully mine fore-background information from templates and search regions, and utilizes them to improve the discriminative ability without attaching additional models.

Visual transformer and attention variants. Nowadays, Transformer has been rapidly applied in the computer vision due to its global interaction ability. Typically, ViT [14] realizes image classification by learning global representations of patches through attention mechanisms. To optimize efficiency, Swin Transformer [38] restricts attention computation in non-overlapping local windows and exchange information between windows through a sliding operation. Thanks to these effective designs, Vision Transformers have been applied to visual tracking, such as MixFormer [10] (using CvT [55]) and OSTrack [62] (using ViT [14]). Further, some trackers improve attention mechanisms to better adapt to visual tracking task. AiATrack [22] proposes an attention in attention module to model the relationship of attention scores, which can improve the quality of attention maps. And SparseTT [20] proposes sparse attention to focus the most relevant information in the search region. However, these methods still calculate attention scores with appearance similarities, which will inevitably disturbed by similar distractors. Differently, we propose a novel foreground-background distribution modeling tracker, which can accurately discriminate target features by perceiving the distribution of fore-background.

3. Method

In this section, we introduce our proposed foreground-background distribution modeling transformer for visual tracking (F-BDMTrack). The architecture is in Figure 2.

3.1. Overview

As illustrated in Figure 2, our proposed tracker consists of L stacked fore-background distribution modeling transformer blocks, where each block have two key components including the fore-background agent learning (FBAL) module and the distribution-aware attention (DA^2) module. Given a pair of images including template image $\mathbf{z} \in \mathbb{R}^{3\times H_z\times W_z}$ and search region image $\mathbf{x} \in \mathbb{R}^{H_x\times W_x\times 3}$, we first split and flatten them to obtain the patch sequence of template $\mathbf{z}_p \in \mathbb{R}^{N_z\times(3\cdot p^2)}$ and search region $\mathbf{x}_p \in \mathbb{R}^{N_x\times(3\cdot p^2)}$, where (p,p) is the patch resolution, and $N_z = (H_zW_z)/p^2, N_x = (H_xW_x)/p^2$ are the number of patches for template and search region respectively. Similar to plain vision transformers [14], these patch sequences are mapped into C dimensions by a linear projection, and learnable position embeddings are added to obtain template token embeddings $\mathbf{H}_z^0 \in \mathbb{R}^{N_z\times C}$ and search region token embeddings $\mathbf{H}_x^0 \in \mathbb{R}^{N_x\times C}$. Then, $\mathbf{H}_z^0$ and $\mathbf{H}_x^0$ are concatenated as the input ($\mathbf{H}_{zx}^0 = [\mathbf{H}_z^0; \mathbf{H}_x^0]$) to the fore-background distribution modeling transformer block. In this transformer block, we intergraded two modules (the FBAL module and the DA^2 module) to help learn discriminative features for both search region and template (We will give a detailed introduction about how these two modules work in our proposed transformer block below). Here, we denote $\mathbf{H}_{zx}^l = [\mathbf{H}_z^l; \mathbf{H}_x^l]$ as token embeddings output from the l^{th} transformer block. Eventually, we reshape search region token features $\mathbf{H}_x^L$ from the last block, and send them into a box prediction head to estimate the target state.

3.2. Fore-background Agent Learning Module

Given token features $[\mathbf{H}_z^{l-1}; \mathbf{H}_x^{l-1}]$ from $(l-1)^{th}$ transformer block, we normalize them with Layer Normalization [1] to obtain $\widehat{\mathbf{H}}_z^l$=LN($\mathbf{H}_z^{l-1}$), and $\widehat{\mathbf{H}}_x^l$=LN($\mathbf{H}_x^{l-1}$). These normalized features $\widehat{\mathbf{H}}_z^l$ and $\widehat{\mathbf{H}}_x^l$ are sent into the fore-background agent learning (FBAL) module to produce fore-background agent (FB-agent) for the template and search region. Specifically, $\widehat{\mathbf{H}}_z^l$ and $\widehat{\mathbf{H}}_x^l$ are first projected into C_e dimensions ($C_e \ll C$) to obtain $\mathbf{E}_z^l$ and $\mathbf{E}_x^l$, which can reduce computational overhead. Then, we design a pseudo bounding box (bbox) generation strategy to obtain

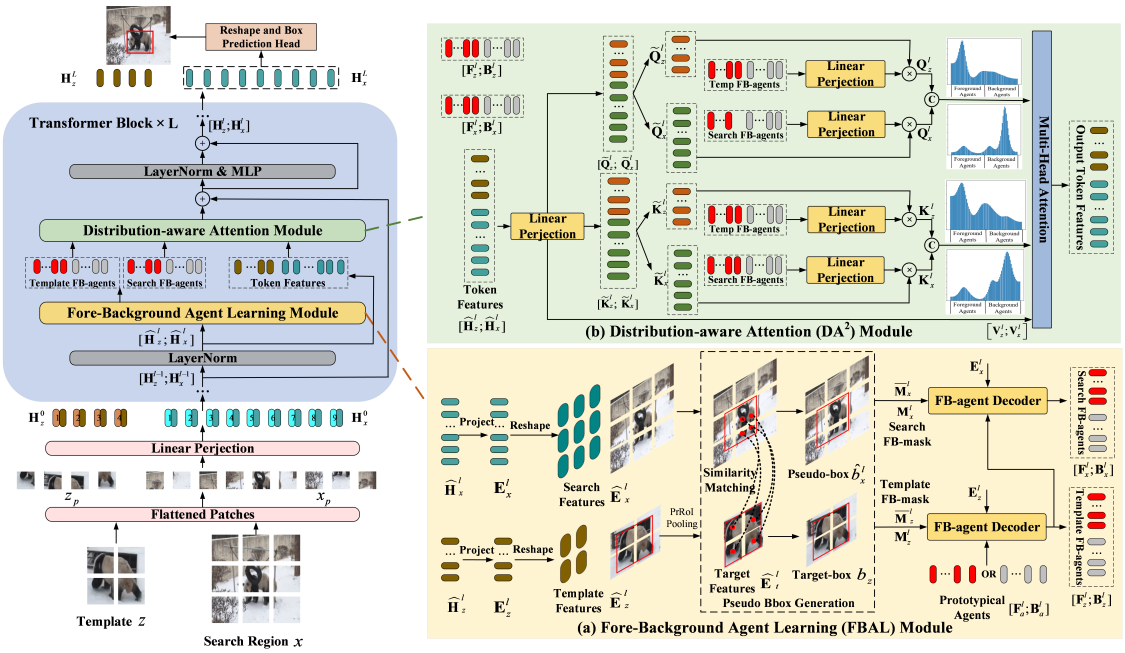

Figure 2. The architecture of our F-BDMTrack consists of L stacked fore-background distribution modeling transformer blocks, where each block have two key components including the fore-background agent learning (FBAL) module and the distribution-aware attention (DA^2) module. The FBAL module is used to produce fore-background agents for the template (template FB-agents) and the search region (search FB-agents). And the DA^2 module is used for feature aggregation by modeling the fore-background distribution. The output from the l^{th} transformer block is denoted as $[\mathbf{H}_z^l; \mathbf{H}_x^l]$. Finally, we reshape search region token features $\mathbf{H}_x^L$ from the last block, and send them into a box prediction head to estimate the target state. For more details, please refer to the text.

the bounding box for the template and search region. Finally, we can obtain fore-background agents (FB-agents) for the template and search region by leveraging a FB-agent decoder to aggregate information within or outside the bounding box. The pseudo bbox generation strategy and the FB-agent decoder are introduced as follows.

Pseudo bbox generation. Since the ground-truth bbox for the template is given in advance, the difficulty is how to generate pseudo bbox for the search region. To this end, we establish coarse matches between template and search region, which can help generate pseudo bbox. Specifically, given $\mathbf{E}_z^l$ and $\mathbf{E}_x^l$, we first reshape them into $\widehat{\mathbf{E}}_z^l \in \mathbb{R}^{\frac{H_z}{p} \times \frac{W_z}{p} \times C_e}$, and $\widehat{\mathbf{E}}_x^l \in \mathbb{R}^{\frac{H_x}{p} \times \frac{W_x}{p} \times C_e}$. Then the target feature $\widehat{\mathbf{E}}_t^l \in \mathbb{R}^{h_t \times w_t \times C_e}$ is cropped according to the ground-truth bbox b_z, *i.e.* $\widehat{\mathbf{E}}_t^l = \mathrm{PrPool}(\widehat{\mathbf{E}}_z^l, b_z)$. Here, $\mathrm{PrPool}(\cdot,\cdot)$ denotes the Precise RoI Pooling [28]. Next, we conduct similarity matching to compute the matching point in $\widehat{\mathbf{E}}_x^l$ for each point in $\widehat{\mathbf{E}}_t^l$ according to the probability $\boldsymbol{D}^l$. Formally,

$$\begin{aligned} \boldsymbol{D}^l(k,i) &= \frac{\exp(< \widehat{\mathbf{E}}_t^l(k), \widehat{\mathbf{E}}_x^l(i) >)}{\sum_{i=1}^{N_x} \exp(< \widehat{\mathbf{E}}_t^l(k), \widehat{\mathbf{E}}_x^l(i) >)}, \\ (\hat{x}_k^l, \hat{y}_k^l) &= \sum_{i=1}^{N_x} s \cdot (x_i, y_i) \cdot \boldsymbol{D}^l(k,i), \end{aligned} \tag{1}$$

where $k = (x_k, y_k)$ and $i = (x_i, y_i)$ enumerate all 2D positions in $\widehat{\mathbf{E}}_t^l$ and $\widehat{\mathbf{E}}_x^l$, respectively. s is the stride of the backbone network. In this way, we can obtain a set of matching points $\{(\hat{x}_k^l, \hat{y}_k^l)\}_{k=1}^K$, where $K = h_t w_t$. Finally, we derive the pseudo bbox $\hat{b}_x^l = (\hat{x}^l, \hat{y}^l, \hat{w}^l, \hat{h}^l)$ with the mean and the standard deviation of all keypoints inspired by RepPoint [60]. Please refer to the **Supplementary Material** for details to obtain $\hat{b}_x^l$. During training, we use ground-truth bbox b_x of the search region as supervision for more accurate pseudo bbox generation. In specific, a set of keypoints $\{(x_j, y_j)\}_{j=1}^K$ are uniformly sampled within the bbox b_x. And the point loss $\mathcal{L}_{point}$ measured by Chamfer distance [18, 61] is introduced to constrain the predicted points $\{(\hat{x}_k^l, \hat{y}_k^l)\}_{k=1}^K$ scattered in the bbox b_x.

$$\begin{aligned} \mathcal{L}_{point}^l = &\frac{1}{K}\sum\nolimits_{j=1}^{K} \min_k \left\|(x_j, y_j) - (\hat{x}_k^l, \hat{y}_k^l)\right\|_2 \\ &+ \frac{1}{K}\sum\nolimits_{k=1}^{K} \min_j \left\|(x_j, y_j) - (\hat{x}_k^l, \hat{y}_k^l)\right\|_2 . \end{aligned} \tag{2}$$

Besides, we introduce ℓ_1 loss and generalized IoU loss [48] to directly constrain predicted pseudo bbox. And the final pseudo bbox generation loss is as follows.

$$\mathcal{L}_{box}^l = \mathcal{L}_{point}^l + \mathcal{L}_1^l(b_x, \hat{b}_x^l) + \mathcal{L}_{giou}^l(b_x, \hat{b}_x^l). \tag{3}$$

FB-agent decoder. After obtaining the bbox for the template b_z and search region $\hat{b}_x^l$, we aim to aggregate information within or outside the bbox. To this end, we first initialize a set of prototypical foreground agents $\mathbf{F}_a^l \in \mathbb{R}^{N_a \times C_e}$

and prototypical background agents $\mathbf{B}_a^l \in \mathbb{R}^{N_a \times C_e}$. Inspired by CrossViT [6], we use masked cross-attention (M-CA) mechanism to aggregate information of $\mathbf{E}_z^l$ in the bbox b_z, and produce template foreground agents. Formally,

$$\begin{aligned} \mathbf{A}_{tt} &= \text{Softmax}((\mathbf{F}_a^l \mathbf{W}_Q)(\mathbf{E}_z^l \mathbf{W}_K)^\top + \mathbf{M}_z^l), \\ \mathbf{F}_z^l &= \text{M-CA}(\mathbf{F}_a^l, \mathbf{E}_z^l, \mathbf{M}_z^l) = \mathbf{A}_{tt}(\mathbf{E}_z^l \mathbf{W}_V), \end{aligned} \tag{4}$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are linear projections. And the element in template foreground mask $\mathbf{M}_z^l$ is set to 0, if the corresponding position of this element is in the bbox b_z. Otherwise, it will be set to $-\infty$. Similarly, for the template background mask $\overline{\mathbf{M}}_z^l$, the element is set to 0, if the corresponding position of this element is outside the bbox b_z. Otherwise, it will be set to $-\infty$. And we can obtain template background agents $\mathbf{B}_z^l = \text{M-CA}(\mathbf{B}_a^l, \mathbf{E}_z^l, \overline{\mathbf{M}}_z^l)$.

For search region features $\mathbf{E}_x^l$, we can generate $\mathbf{M}_x^l$ and $\overline{\mathbf{M}}_x^l$ according to the pseudo bbox $\hat{b}_x^l$. Since $\hat{b}_x^l$ is not precise enough, we leverage updated template FB-agents $[\mathbf{F}_z^l; \mathbf{B}_z^l]$ to serve as queries, which provides clearer fore-background priors than prototypical agents $[\mathbf{F}_a^l; \mathbf{B}_a^l]$. Formally,

$$\begin{aligned} \mathbf{F}_x^l &= \text{M-CA}(\mathbf{F}_z^l, \mathbf{E}_x^l, \mathbf{M}_x^l), \\ \mathbf{B}_x^l &= \text{M-CA}(\mathbf{B}_z^l, \mathbf{E}_x^l, \overline{\mathbf{M}}_x^l). \end{aligned} \tag{5}$$

Finally, we obtain template FB-agents $[\mathbf{F}_z^l; \mathbf{B}_z^l]$, and search region FB-agents $[\mathbf{F}_x^l; \mathbf{B}_x^l]$ at the l^{th} block.

3.3. Distribution-aware Attention Module

After obtaining template FB-agents $[\mathbf{F}_z^l; \mathbf{B}_z^l]$ and search region FB-agents $[\mathbf{F}_x^l; \mathbf{B}_x^l]$, we begin to model the fore-background distribution for each image token. Specifically, given token features $[\mathbf{H}_z^{l-1}; \mathbf{H}_x^{l-1}]$, the initial query, key and value arise from these token features. Formally,

$$\begin{aligned} \widetilde{\mathbf{Q}}_{zx}^l &= [\widetilde{\mathbf{Q}}_z^l; \widetilde{\mathbf{Q}}_x^l] = \text{LN}([\mathbf{H}_z^{l-1}; \mathbf{H}_x^{l-1}])\mathbf{W}_Q^l, \\ \widetilde{\mathbf{K}}_{zx}^l &= [\widetilde{\mathbf{K}}_z^l; \widetilde{\mathbf{K}}_x^l] = \text{LN}([\mathbf{H}_z^{l-1}; \mathbf{H}_x^{l-1}])\mathbf{W}_K^l, \\ \mathbf{V}_{zx}^l &= [\mathbf{V}_z^l; \mathbf{V}_x^l] = \text{LN}([\mathbf{H}_z^{l-1}; \mathbf{H}_x^{l-1}])\mathbf{W}_V^l, \end{aligned} \tag{6}$$

where LN denotes the layer normalization, $\mathbf{W}_K^l \in \mathbb{R}^{C \times C}, \mathbf{W}_Q^l \in \mathbb{R}^{C \times C}, \mathbf{W}_V^l \in \mathbb{R}^{C \times C}$ are linear projections. As can be seen, existing plain attention mechanism [14] generate queries and keys simply based on appearance features, which will easily have incorrect high attention score for regions with similar distractors. Differently, we use the fore-background distribution to serve as queries and keys instead of appearance features. To this end, we first project $[\mathbf{F}_z^l; \mathbf{B}_z^l]$ and $[\mathbf{F}_x^l; \mathbf{B}_x^l]$ into C dimensions. By calculating the similarity between each token and these fore-background agents, we can model the fore-background distribution of this token. Here, we take the fore-background distribution of initial query token as updated queries $\mathbf{Q}_{zx}^l = [\mathbf{Q}_z^l; \mathbf{Q}_x^l]$. Formally,

$$\begin{aligned} \mathbf{Q}_z^l &= \widetilde{\mathbf{Q}}_z^l [\mathbf{F}_z^l \mathbf{W}_z^l; \mathbf{B}_z^l \mathbf{W}_z^l]^\top, \\ \mathbf{Q}_x^l &= \widetilde{\mathbf{Q}}_x^l [\mathbf{F}_x^l \mathbf{W}_x^l; \mathbf{B}_x^l \mathbf{W}_x^l]^\top, \end{aligned} \tag{7}$$

where $\mathbf{W}_z^l, \mathbf{W}_x^l \in \mathbb{R}^{C_e \times C}$ are linear projections. Similarly, we can obtain updated keys $\mathbf{K}_{zx}^l = [\mathbf{K}_z^l; \mathbf{K}_x^l]$.

$$\begin{aligned} \mathbf{K}_z^l &= \widetilde{\mathbf{K}}_z^l [\mathbf{F}_z^l \mathbf{W}_z^l; \mathbf{B}_z^l \mathbf{W}_z^l]^\top, \\ \mathbf{K}_x^l &= \widetilde{\mathbf{K}}_x^l [\mathbf{F}_x^l \mathbf{W}_x^l; \mathbf{B}_x^l \mathbf{W}_x^l]^\top. \end{aligned} \tag{8}$$

With updated queries and keys, the proposed distribution-aware attention module can achieve fore-background aware feature interaction. The output of the l^{th} transformer blocks $\mathbf{H}_{zx}^l$ can be formulated as follows,

$$\begin{aligned} \widetilde{\mathbf{H}}_{zx}^l &= \text{Softmax}(\frac{\mathbf{Q}_{zx}^l (\mathbf{K}_{zx}^l)^\top}{\sqrt{C}})\mathbf{V}_{zx}^l + \mathbf{H}_{zx}^{l-1} \\ \mathbf{H}_{zx}^l &= [\mathbf{H}_z^l; \mathbf{H}_x^l] = \text{MLP}(\text{LN}(\widetilde{\mathbf{H}}_{zx}^l)) + \widetilde{\mathbf{H}}_{zx}^l, \end{aligned} \tag{9}$$

where MLP denotes the multi-layer perception. The output token features $\mathbf{H}_{zx}^l$ will be sent into next transformer blocks. Eventually, token features are processed L times via our proposed fore-background distribution modeling transformer block, and we can obtain final token features $\mathbf{H}_{zx}^L$.

3.4. Box Prediction Head and Objective Function

As shown in Figure 2, search region token features $\mathbf{H}_x^L$ are reshaped and fed into a box prediction head for target state estimation. Here, the box prediction head is inspired from OSTrack [62]. We use the weighted focal loss [33] as classification loss $\mathcal{L}_{cls}$. Besides, we adopt ℓ_1 loss $\mathcal{L}_1$ and generalized IoU loss [48] $\mathcal{L}_{giou}$ to supervise bounding box regression. More details about the box prediction head and losses can be referred to OSTrack [62]. As we introduced before, there are point loss and box loss for pseudo bbox generation at each block. In summary, the overall objective function is formulated as follows.

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_1 + \lambda_{giou} \mathcal{L}_{giou} + \lambda_{box} \sum_{l=1}^{L} \mathcal{L}_{box}^l, \tag{10}$$

where λ_1, λ_{giou} and λ_{box} are scalars to balance these losses.

4. Experiments

In this section, we first introduce implementation details. Then, we show experimental results on eight benchmarks. Finally, we conduct a series of ablation studies to verify the effectiveness of each component. Please refer to the **Supplementary Material** for more details and results.

Table 1. Comparisons with state-of-the-art trackers on GOT-10k, TNL2K, LaSOT, LaSOT$_{ext}$, and TrackingNet. The best three results are shown in **red**, **blue** and **green** fonts.

Method	GOT-10k [26]			TNL2K [54]		LaSOT [17]			LaSOT$_{ext}$ [16]			TrackingNet [45]		
	AO	SR$_{0.5}$	SR$_{0.75}$	AUC	P	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P
SiamFC[2]	34.8	35.3	9.8	29.5	28.6	33.6	42.0	33.9	23.0	31.1	26.9	57.1	66.3	53.3
ECO [11]	31.6	30.9	11.1	32.6	31.7	32.4	33.8	30.1	22.0	25.2	24.0	55.4	61.8	49.2
SiamRPN++ [34]	51.7	61.6	32.5	41.3	41.2	49.6	56.9	49.1	34.0	41.6	39.6	73.3	80.0	69.4
SiamCAR [23]	56.9	67.0	41.5	35.5	38.4	50.7	60.0	51.0	-	-	-	-	-	-
SiamFC++ [58]	59.5	69.5	47.9	38.6	36.9	54.4	62.3	54.7	-	-	-	75.4	80.0	70.5
D3S [39]	59.7	67.6	46.2	38.8	39.3	-	-	-	-	-	-	72.8	76.8	66.4
Ocean [63]	61.1	72.1	47.3	38.4	37.7	56.0	65.1	56.6	-	-	-	-	-	-
DiMP-50 [3]	61.1	71.7	49.2	44.7	43.4	56.9	65.0	56.7	39.2	47.6	45.1	74.0	80.1	68.7
PrDiMP-50 [13]	63.4	73.8	54.3	47.0	45.9	59.8	68.8	60.8	-	-	-	75.8	81.6	70.4
KeepTrack[43]	-	-	-	-	-	67.1	77.2	70.2	48.2	-	-	-	-	-
ToMP-101 [42]	-	-	-	-	-	68.5	79.2	73.5	45.9	-	-	81.5	86.4	78.9
KYS [4]	63.6	75.1	51.5	44.9	43.5	55.4	63.3	-	-	-	-	74.0	80.0	68.8
STMTrack [21]	64.2	73.7	57.5	-	-	60.6	63.3	69.3	-	-	-	80.3	85.1	76.7
TransT [7]	67.1	76.8	60.9	50.7	51.7	64.9	69.0	73.8	-	-	-	81.4	86.7	80.3
STARK-ST101 [59]	68.8	78.1	64.1	-	-	67.1	77.0	-	-	-	-	82.0	86.9	-
CSWinTT [50]	69.4	78.9	65.4	-	-	66.2	75.2	70.9	-	-	-	81.9	86.7	79.5
SimTrack-B/16 [5]	68.6	78.9	62.4	54.8	**53.8**	69.3	78.5	-	-	-	-	82.3	86.5	-
SwinTrack-B [36]	69.4	78.0	64.3	-	-	69.6	78.6	74.1	47.6	58.2	54.1	82.5	87.0	80.4
AiATrack [22]	69.6	80.0	63.2	-	-	69.0	79.4	73.8	-	-	-	82.7	87.8	80.4
SBT-L [57]	70.4	80.8	64.7	-	-	66.7	77.1	-	-	-	-	-	-	-
MixFormer [10]	70.7	80.0	67.8	-	-	69.2	78.7	74.7	-	-	-	83.1	88.1	81.6
RTS [42]	-	-	-	-	-	69.7	76.2	73.7	-	-	-	81.6	86.0	79.4
OSTrack-256 [62]	71.0	80.4	68.2	54.3	-	69.1	78.7	75.2	47.4	57.3	53.3	83.1	87.8	82.0
OSTrack-384 [62]	**73.7**	**83.2**	**70.8**	**55.9**	-	**71.1**	**81.1**	**77.6**	**50.5**	**61.3**	**57.6**	**83.9**	**88.5**	**83.2**
F-BDMTrack-256	**72.7**	**82.0**	**69.9**	**56.4**	**56.5**	**69.9**	**79.4**	**75.8**	**47.9**	**57.9**	**54.0**	**83.7**	**88.3**	**82.6**
F-BDMTrack-384	**75.4**	**84.3**	**72.9**	**57.8**	**59.4**	**72.0**	**81.5**	**77.7**	**50.8**	**61.3**	**57.8**	**84.5**	**89.0**	**84.0**

4.1. Implementation Details

Network details. We present two variants of our tracker, F-BDMTrack-256 and F-BDMTrack-384. Similar to OSTrack [62], we crop the template image and search region image which are 2^2 and 4^2 times of the target box area respectively. The F-BDMTrack-256 resizes search region to a resolution of 256×256 and template to a resolution of 128×128 as input. The F-BDMTrack-384 resizes search region to a resolution of 384×384 and template to 192×192. These images are split into a set of patches, where the resolution of each patch is 16×16. These patches are flattened and projected into $C = 768$ channels to serve as the input of F-BDMTrack. The channel of $\mathbf{E}_z^l$ in the FBAL module is set to $C_e = 256$. We leverage the Precise RoI Pooling [28] operation to crop features according to the target bounding box, and obtain the target feature with shape $(h_t, w_t) = (4, 4)$. The number of foreground agents and background agents are both set to $N_a = 4$. And our F-BDMTrack consists of $L = 12$ fore-background distribution modeling transformer blocks.

Training details. We train our model on the training splits of LaSOT [17], GOT-10K [26], COCO2017 [37], and TrackingNet [45], which is a similar training setting to OSTrack [62]. Common data augmentations including horizontal flip and brightness jittering are applied in the training process. We train our F-BDMTrack by the AdamW optimizer with the weight decay 10^{-4}. The learning rates start from 4×10^{-5} for the backbone including the FBAL module and the DA^2 module, and 4×10^{-4} for the box prediction head. Our model is trained on four NVIDIA RTX 3090 GPUs, each GPU holds 32 image pairs, resulting in a total batch size of 128. The total epochs are set to 300 with 60k samples per epoch and we decrease the learning rate by a factor of 10 after 240 epochs. The weights of training losses are set to $\lambda_1 = 5.0$, $\lambda_{giou} = 2.0$, and $\lambda_{box} = 1.0$.

4.2. Results and State-of-the-art Comparisons

GOT-10k. GOT-10k [26] is a large-scale dataset containing over 10k videos for training and 180 for testing. It forbids trackers to use external datasets for training. We follow its policy and retrain our model. The results are reported in Table 1. Our tracker has a satisfactory improvement in all metrics. Specifically, F-BDMTrack-256 improves by 1.7% in success rate (SR$_{0.75}$) compared with OSTrack-256. And F-BDMTrack-384 gives an improved new record 75.4% in AO, indicating the effectiveness of foreground and background distribution modeling to extract discriminative features, resulting in more accurate state estimation.

TNL2K. TNL2K [54] is a recently published dataset containing 700 video sequences for testing. In Table 1, F-BDMTrack-256 gains by 2.1% AUC over OTrack-256 and

Table 2. Comparisons with state-of-the-art trackers on NFS, OTB100 and UAV123 datasets in terms of overall AUC score. The best three results are shown in **red**, **blue** and **green** fonts.

	SiamFC [2]	Ocean [63]	ATOM [12]	DiMP-50 [3]	PrDiMP-50 [13]	TransT [7]	OSTrack-256 [62]	OSTrack-384 [62]	F-BDMTrack-256	F-BDMTrack-384
NFS [30]	37.7	49.4	58.3	61.8	63.5	65.3	64.7	**66.5**	**66.0**	**67.3**
OTB100 [56]	61.2	68.4	66.3	68.4	**69.6**	69.4	68.1	69.2	**69.5**	**69.9**
UAV123 [44]	46.8	57.4	63.2	64.3	68.0	68.1	68.3	**70.7**	**69.0**	**70.9**

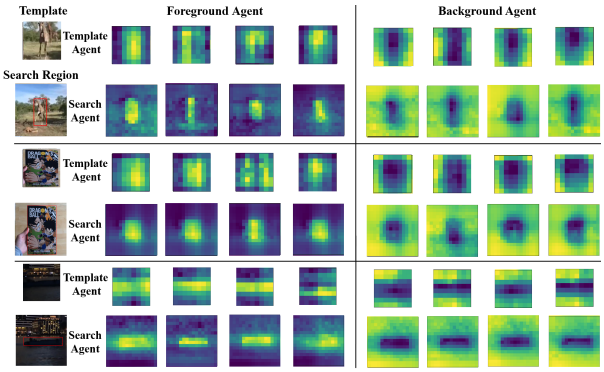

Figure 3. Visualizations of fore-background agent activation maps for the template and search region.

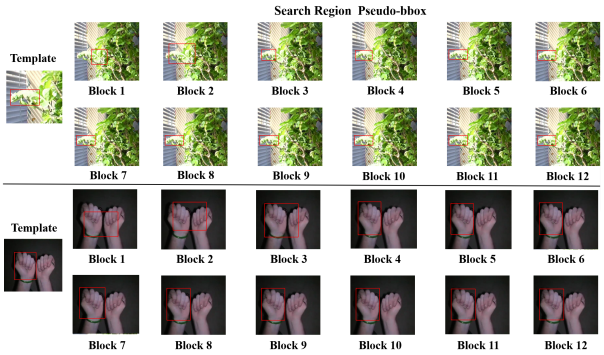

Figure 4. Visualization of pseudo bboxes using the pseudo-bbox generation strategy at all blocks.

Table 3. Effectiveness of each component on the LaSOT

Model	FBAL	DA^2	AUC	P_{Norm}	P
[A]	✗	✗	68.5	77.8	74.2
[B]	✓	✗	68.8	78.1	74.6
[C]	✗	✓	69.4	78.7	75.4
[D]	✓	✓	**69.9**	**79.4**	**75.8**

even performances better than larger models (OTrack-384). When also leveraging a large model (F-BDMTrack-384), we can further improve the AUC score to 57.8%.

LaSOT. LaSOT [17] is a densely annotated large-scale dataset containing a total of 1400 long-term video sequences, of which the test set contains 280 video sequences. This is a challenging tracking dataset. As shown in Table 1, F-BDMTrack-256 outperforms OSTrack-256 by 0.8% AUC. Further, F-BDMTrack-384 achieves the best performance of 72.0% AUC. Our tracker does not add additional timing strategies [22, 10, 59] in this long-term tracking task, but can still obtain performance improvement with the aid of fore-background distribution modeling design, which demonstrates that our approach can fundamentally mitigate complex scenarios and distractors.

LaSOT$_{ext}$. The recent LaSOT$_{ext}$ [16] is an extended subset of LaSOT, containing 150 additional new sequences. This dataset has many similar distractors, making it difficult for tracking. And our F-BDMTrack can achieve competitive or even better performance. In specific, F-BDMTrack-256 achieves superior results, outperforming OSTrack-256 by 0.5% AUC. And F-BDMTrack-356 obtains a much higher performance with 50.8% AUC.

Trackingnet. Trackingnet [45] provides over 30k video sequences, which are sampled from Youtube to cover target categories and scenes of real-world. It provides 511 testing video sequences without publicly available annotation, so results reported in Table 1 are obtained from the online evaluation server. As can be seen, our F-BDMTrack-384 achieves 84.5% AUC and sets a new state of the art.

NFS. Videos in Need for Speed(NFS) [30] are captured from a high frame rate camera, which contains fast motions and distractors. We report results on its commonly used version NFS30. As shown in Table 2, F-BDMTrack-256 achieves 66.0% AUC, outperforming OSTrack-256 by 1.3%. And F-BDMTrack-384 achieves best results (67.3% AUC score) as expected, demonstrating the superior competitiveness of our tracker in the presence of interferer-challenged benchmark.

OTB100. OTB [56] is a pioneering visual tracking benchmark. It has been noticed that this benchmark is approaching saturation [53, 59, 43]. In Table 2, two versions of our tracker both achieve promising AUC scores (69.5 and 69.9).

UAV123. UAV123 [44] provides 123 unmanned aerial vehicle(UAV) sequences, including different scale sizes of UAV in real-world dynamic scenarios. As shown in Table 2, our tracker achieves the best AUC score (70.9%) and is suitable for UAV tracking scenarios.

4.3. Ablation Study

To validate the effectiveness of each component, we perform a detailed ablation study on LaSOT [17]. The following experiments use F-BDMTrack-256 as the base model.

Effectiveness of key components in our F-BDMTrack. Our tracker consists of two key components, including a fore-background agent learning (FBAL) module and a distribution-aware attention (DA^2) module. Here, we explore the exact impact of these modules. The model [A] is the baseline, which means that we directly use plain attention mechanism without using the FBAL and the DA^2 module. For the model [B], we add the FBAL module, and

Table 4. Effects with different numbers of prototypical FB-agents.

N_a	AUC	P_{Norm}	P
1	69.3	78.8	75.5
2	69.6	79.1	75.6
4	**69.9**	**79.4**	**75.8**
8	69.3	78.9	75.2
16	69.0	78.6	75.1

Table 5. Effects of the pseudo bbox generation at different blocks.

blocks	AUC	P_{Norm}	P
None	69.0	78.5	75.0
10-12	69.5	79.1	75.5
6-12	**69.9**	**79.4**	**75.8**
4-12	69.8	79.2	75.8
1-12	69.2	78.7	75.2

adopt plain attention mechanism to replace our designed DA^2 module. The plain attention mechanism does not require fore-background agents (FB-agents) produced by our FBAL module. Thus, the model [B] is used to validate the impacts of pseudo bbox generation loss $\mathcal{L}_{box}^{l}$ introduced in the FBAL module. As shown in Table 3, the performance of model [B] gains by 0.3% compared to the model [A], demonstrating that the pseudo bbox loss has some positive effects to constrain foreground features interact with features in foreground regions. For the model [C], we adopt the DA^2 module for feature interaction, and FB-agents are obtained by pooling features within or outside the bounding box for the template and search region. As can be seen, the performance of model [C] gains by 0.9% compared to the model [A], which demonstrates that the proposed DA^2 module can significantly help learn discriminative features even if FB-agents are not good enough. When adding the FBAL module for better FB-agents generation, the performance of model [D] gains by 0.5% compared to the model [C], showing the superiority of proposed FBAL module.

Study on the number of prototypical FB-agents in the FBAL module. Here, we explore the effects about different numbers of prototypical FB-agents. As shown in Table 4, our tracker achieves the optimal result (69.9% AUC score) when $N_a = 4$. A small number of prototypical FB-agents cannot fully model the fore-background distribution, leading to decreased performance. Meanwhile, more prototypical FB-agents will introduce useless noise, which is harmful for feature interaction. Besides, without explicit constraints to guide foreground and background agent learning, an excessive number of agents are unable to focus on their respective discriminative information better. A more effective FB-agent learning way can be explored in future work. Finally, we show the fore-background agent activation maps for the template and search region in Figure 3. It can be seen that these FB-agents generated by our FBAL module can focus on the foreground and background well.

Study on the pseudo bbox generation strategy at different blocks. High-quality pseudo bbox is more conducive to the FB-agent learning for the search region. However, as shown in Figure 4, the pseudo-bboxs in shallow blocks tend to have a large bias, which is harmful to learn effective FB-agents. We conduct a detailed experiment to find which blocks are optimal for using pseudo bbox generation strategy. As shown in Table 5, our tracker achieves the best results when we leverage the pseudo bbox generation strategy at the 6^{th} to 12^{th} ('6-12') transformer blocks. This is because the last few blocks provide more reliable pseudo bbox to guide fore-background agent learning.

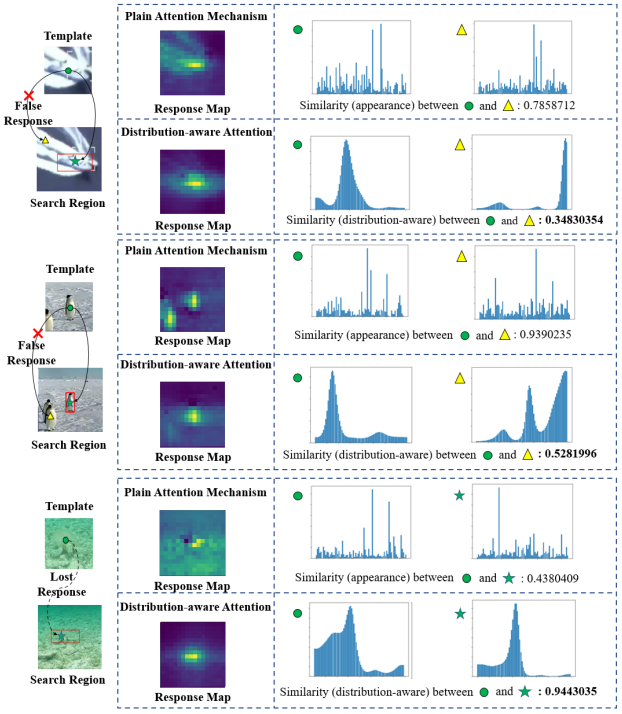

Figure 5. Response maps of the template central feature to all search region features. It can be seen that although the template target has similar appearance with the background distractor in the search region, their fore-background distributions are different.

Comparisons between the plain attention mechanism and our DA^2 mechanism. Here, we visualize response maps of the plain attention mechanism and our DA^2 mechanism in Figure 5. As we can see, the plain attention mechanism easily focus on complex backgrounds or distractors. This is because attention scores are obtained based on appearance similarities, while targets usually have similar appearance with distractors, resulting in incorrect response map. Differently, our proposed DA^2 mechanism discriminates features from the perspective of fore-background distributions instead of appearance similarities, which can effectively suppress the response of the template to background distractors in the search region, and strengthen the response of the template to the foreground target in the search region. Finally, based on the proposed distribution-

aware attention mechanism, our tracker can achieve superior performance on all eight tracking benchmarks.

5. Conclusion

In this work, we propose a novel foreground-background distribution modeling transformer for visual tracking, including a FBAL module and a DA^2 module. With these two elegant designs, our proposed tracker can extract features with high target discriminative power under ever-changing tracking scenarios, which is essential for accurate target state estimation. Extensive experiments on eight tracking benchmarks verify the superiority of our proposed tracker.

6. Acknowledgement

This work was partially supported by the National Nature Science Foundation of China (Grant 62022078, 62021001), and National Defense Basic Scientific Research Program of China (Grant JCKY2020903B002).

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[2] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip H S Torr. Fully-convolutional siamese networks for object tracking. In *Proceedings of the European Conference on Computer Vision Workshops*, 2016. 1, 2, 6, 7

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 1, 2, 6, 7

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know Your Surroundings: Exploiting scene information for object tracking. In *Proceedings of the European Conference on Computer Vision*, 2020. 6

[5] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qiuhong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: A simplified architecture for visual object tracking. In *Proceedings of the European Conference on Computer Vision*, 2022. 1, 2, 3, 6

[6] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021. 5

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 6, 7

[8] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[9] Linsong Cheng, Jiliang Wang, and Yinghui Li. Vitrack: Efficient tracking on the edge for commodity video surveillance systems. *IEEE Transactions on Parallel and Distributed Systems*, 33(3):723–735, 2021. 1

[10] Yutao Cui, Jiang Cheng, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 3, 6, 7

[11] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient convolution operators for tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6

[12] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate tracking by overlap maximization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 7

[13] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 6, 7

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021. 2, 3, 5

[15] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021. 1

[16] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Mingzhen Huang, Juehuan Liu, Yong Xu, et al. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129(2):439–461, 2021. 6, 7

[17] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 6, 7

[18] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 4

[19] Qi Feng, Vitaly Ablavsky, Qinxun Bai, and Stan Sclaroff. Siamese natural language tracker: Tracking by natural language descriptions with siamese trackers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5851–5860, 2021. 3

[20] Zhihong Fu, Zehua Fu, Qingjie Liu, Wenrui Cai, and Yunhong Wang. Sparsett: Visual tracking with sparse transformers. *arXiv preprint arXiv:2205.03776*, 2022. 3

[21] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time

memory networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13774–13783, 2021. 2, 3, 6

[22] Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. Aiatrack: Attention in attention for transformer visual tracking. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 146–164. Springer, 2022. 2, 3, 6, 7

[23] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 6

[24] Qing Guo, Ziyi Cheng, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Yang Liu, and Jianjun Zhao. Learning to adversarially blur visual object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10839–10848, 2021. 1

[25] Bo Huang, Junjie Chen, Tingfa Xu, Ying Wang, Shenwang Jiang, Yuncheng Wang, Lei Wang, and Jianan Li. Siamsta: Spatio-temporal attention based siamese tracker for tracking uavs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1204–1212, 2021. 1

[26] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 6

[27] Irene Anindaputri Iswanto, Tan William Choa, and Bin Li. Object tracking based on meanshift and particle-kalman filter algorithm with multi features. *Procedia computer science*, 157:521–529, 2019. 1

[28] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision*, 2018. 4, 6

[29] Nan Jiang, Kuiran Wang, Xiaoke Peng, Xuehui Yu, Qiang Wang, Junliang Xing, Guorong Li, Qixiang Ye, Jianbin Jiao, Zhenjun Han, et al. Anti-uav: a large-scale benchmark for vision-based uav tracking. *IEEE Transactions on Multimedia*, 2021. 1

[30] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 7

[31] Matej Kristan, Jiří Matas, Aleš Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Hyung Jin Chang, Martin Danelljan, Luka Cehovin, Alan Lukežič, et al. The ninth visual object tracking vot2021 challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2711–2738, 2021. 1

[32] Jin-Peng Lan, Zhi-Qi Cheng, Jun-Yan He, Chenyang Li, Bin Luo, Xu Bao, Wangmeng Xiang, Yifeng Geng, and Xuansong Xie. Procontext: Exploring progressive context transformer for tracking. *arXiv preprint arXiv:2210.15511*, 2022. 1, 2

[33] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 5

[34] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 6

[35] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2

[36] Liting Lin, Heng Fan, Yong Xu, and Haibin Ling. Swintrack: A simple and strong baseline for transformer tracking. *Advances of Neural Information Processing Systems*, 2021. 1, 2, 6

[37] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014. 6

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 3

[39] Alan Lukezic, Jiri Matas, and Matej Kristan. D3S - A discriminative single shot segmentation tracker. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 6

[40] Ziang Ma, Linyuan Wang, Haitao Zhang, Wei Lu, and Jun Yin. Rpt: Learning point set representation for siamese visual tracking. *arXiv preprint arXiv:2008.03467*, 2020. 2

[41] Gayatri Sasi Rekha Machiraju, K Aruna Kumari, and Shaikh Khadar Sharif. Object detection and tracking for community surveillance using transfer learning. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pages 1035–1042. IEEE, 2021. 1

[42] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool. Transforming model prediction for tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8731–8740, 2022. 1, 2, 3, 6

[43] Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 13444–13454, 2021. 6, 7

[44] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In *Proceedings of the European Conference on Computer Vision*, 2016. 7

[45] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision*, 2018. 6, 7

[46] Matthieu Paul, Martin Danelljan, Christoph Mayer, and Luc Van Gool. Robust visual tracking by segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 571–588. Springer, 2022. 3

[47] Chinthaka Premachandra, Shohei Ueda, and Yuya Suzuki. Detection and tracking of moving objects at road intersections using a 360-degree camera for driver assistance and automated driving. *IEEE Access*, 8:135652–135660, 2020. 1

[48] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4, 5

[49] Ahsan Shehzed, Ahmad Jalal, and Kibum Kim. Multi-person tracking in smart surveillance system for crowd counting and normal/abnormal events detection. In *2019 international conference on applied and engineering mathematics (ICAEM)*, pages 163–168. IEEE, 2019. 1

[50] Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Transformer tracking with cyclic shifting window attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8791–8800, 2022. 1, 2, 6

[51] Janani Thangavel, Thanikasalam Kokul, Amirthalingam Ramanan, and Subha Fernando. Transformers in single object tracking: An experimental survey. *arXiv preprint arXiv:2302.11867*, 2023. 1

[52] Paul Voigtlaender, Jonathon Luiten, Philip H. S. Torr, and Bastian Leibe. Siam R-CNN: Visual tracking by redetection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[53] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1571–1580, 2021. 2, 7

[54] Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 13763–13773, 2021. 6

[55] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 22–31, 2021. 2, 3

[56] Yi Wu, Jongwoo Lim, and Ming Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 1, 7

[57] Fei Xie, Chunyu Wang, Guangting Wang, Yue Cao, Wankou Yang, and Wenjun Zeng. Correlation-aware deep tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8751–8760, 2022. 1, 2, 6

[58] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 1, 2, 6

[59] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10448–10457, 2021. 1, 2, 6, 7

[60] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9657–9666, 2019. 4

[61] Ze Yang, Yinghao Xu, Han Xue, Zheng Zhang, Raquel Urtasun, Liwei Wang, Stephen Lin, and Han Hu. Dense reppoints: Representing visual objects with dense point sets. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 227–244. Springer, 2020. 4

[62] Botao Ye, Hong Chang, Bingpeng Ma, and Shiguang Shan. Joint feature learning and relation modeling for tracking: A one-stream framework. *Proceedings of the European Conference on Computer Vision*, 2022. 1, 2, 3, 5, 6, 7

[63] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *Proceedings of the European Conference on Computer Vision*, 2020. 6, 7

[64] Jie Zhao, Jingshu Zhang, Dongdong Li, and Dong Wang. Vision-based anti-uav detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):25323–25334, 2022. 1